AD-A035 793 NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA AIR VEHICL--ETC F/G 9/2
TREAD/TLOOK - A MULTIPURPOSE COMPUTER ROUTINE FOR INTERPOLATION--ETC(U)
JAN 77 M J CADDY
UNCLASSIFIED NADC-76366-30 NL

1 OF 1
ADA035793

END
DATE
FILMED
3-77

REPORT NO. NADC-76366-30

(12)
NW

ADA035793

TREAD/TLOOK - MULTIPURPOSE COMPUTER ROUTINE FOR INTERPOLATION AND EXTRAPOLATION OF TABULAR DATA

M. J. Caddy
Air Vehicle Technology Department
NAVAL AIR DEVELOPMENT CENTER
Warminster, Pennsylvania 18974

11 JANUARY 1977

FINAL REPORT
AIRTASK NO. A5365360/202F/0W45430000
Work Unit No. ORS-1-24

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared for
NAVAL AIR SYSTEMS COMMAND
Department of the Navy
Washington, D.C. 20361

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Department responsible for the report. For example: Report No. NADC-76015-40 indicates the fifteenth Center report for the year 1976, and prepared by the Crew Systems Department. The numerical codes are as follows:

CODE	OFFICE OR DEPARTMENT
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Program and Financial Management Department
07	V/STOL Program Office
09	Technology Management Office
10	Naval Air Facility, Warminster
20	Aero Electronic Technology Department
30	Air Vehicle Technology Department
40	Crew Systems Department
50	Systems Department
60	Naval Navigation Laboratory
81	Technical Support Department
85	Computer Department

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: P. M. Sturm DATE: 11 January 1977

P. M. STURM
Commander, USN
Deputy Director, AVTD

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
14 NADC-76366-30

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9

4. TITLE (and Subtitle)
6 TREAD/TLOOK - A MULTIPURPOSE COMPUTER ROUTINE FOR INTERPOLATION AND EXTRAPOLATION OF TABULAR DATA.

5. TYPE OF REPORT & PERIOD COVERED
Final Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 M. J. Caddy

8. CONTRACT OR GRANT NUMBER(s)
16 W45430
17 W45430000

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Vehicle Technology Department (Code 3013)
Naval Air Development Center
Warminster, PA 18974

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
AIRTASK ~~A5365360/202E/~~0W4543 0000, Work Unit OR6-1-24

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Air Systems Command
Department of the Navy
Washington, D. C. 20361

12. REPORT DATE
11 11 January 1977

13. NUMBER OF PAGES
47
12 45 p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Table Look-up Code
Cubic Spline Fit

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A general purpose computer code is described which permits a rapid, efficient table look-up of one dependent variable in terms of one, two, or three independent variables. A cubic spline fit is used to interpolate or extrapolate between inputted points. The resulting dependent variable function is smooth and continuous in both first and second derivatives.

407 207

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NADC-76366-30

THIS PAGE INTENTIONALLY LEFT BLANK

SUMMARY

INTRODUCTION

Computer programs often require a means of expressing functional relationships in which no exact mathematical expressions are known. In this situation tabular data inputs can be used for these functional relationships. Numerous techniques are available for interrogating table data arrays, from the simplicity of linear technique to the complexity of higher order polynominal techniques. The purpose of this report is to describe a complete computer routine package, TLOOK/TREAD, for input and interpolation of tabular data which is accurate and has broad application to both the vehicle design and technology programs at the Naval Air Development Center as well as other programs which require table look-up information. The computer code can determine the value of a dependent variable in terms of up to three independent variables using a cubic spline interpolation. The inputs to this code require an effective means of locating and correcting input errors. Another report describing a companion computer code, GPPR (General Purpose Plotting Routine), which produces CALCOMP plots of tabular data processed by TLOOK/TREAD will shortly be published as reference (a).

SUMMARY OF RESULTS

A general purpose table look-up cubic spline interpolation technique has been programmed for the CDC 6600 computer. The utilization of the resulting package of routines (called TREAD/TLOOK and SPLNQ1) is illustrated with a sample problem. A comparison between actual and interpolated values of dependent variables in the sample problem has shown that this interpolation technique is a fast, accurate and a valuable computer tool. The programmed cubic spline interpolation technique used by TREAD/TLOOK is greatly improved over that shown in reference (b). The major improvements concern computational speed, computer storage size and flexibility. The input format to TREAD/TLOOK is arranged to accept both symmetrical and nonsymmetrical tabular data arrays.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISCUSSION

BACKGROUND

Numerous codes in current use at NAVAIRDEVCEN require interpolation schemes as a means of expressing functional relationships between dependent and independent variables. Linear and higher order polynominal curve fits represent two approaches for this purpose. Linear interpolation schemes have the advantage of simplicity and fast computational speed. A search procedure is used to determine the correct data point interval where the interpolation is performed. However, unless a large number of data points is input, interpolation accuracy may be lost. In addition, the first derivative of the linear interpolated function is discontinuous at each input data point. Higher order polynominal interpolation schemes exhibit some of the same inaccuracies as linear interpolation schemes. The function obtained from the interpolation may or may not coincide with each input data point. In addition, curve fits tend to oscillate about the data points resulting in numerous inflection points. However, this approach has the advantage of eliminating the search procedure necessary for linear interpolation schemes. The approach suggested in this report circumvents some of these disadvantages.

PROGRAM PACKAGE DESCRIPTION

The table look-up program package in this report consists of the TREAD/TLOOK routine and function SPLNQ1. TREAD/TLOOK is the routine by which tabular data is read as input and stored for the spline fit interpolation by the SPLNQ1 routine. A user's guide in Appendix A illustrates the use of the table look-up package. A general description and operation of the TREAD/TLOOK and SPLNQ1 routines follows.

Description of Subroutine TREAD/TLOOK

Subroutine TREAD (with entry TLOOK) is programmed to interpolate or extrapolate a given set of data with one, two, or three independent variables. The actual interpolation or extrapolation is performed by the function SPLNQ1 and will be discussed later. The TREAD portion of the subroutine reads input data and assembles the data in the format required for function SPLNQ1. The TREAD/TLOOK subroutine has the capability to accept up to 30 sets of data, each set representing a table look-up function of up to three independent variables. The user initiates a TREAD operation to input table data. As the input data is processed and stored in a single dimension array (called TDATA) the starting array location and table reference number of each table is also stored. The table reference number is a 1 to 4 digit number used to identify each table. To use a particular table look-up function the user merely specifies the appropriate table reference number and the values of the independent variables in a call to the TLOOK routine. When a TLOOK operation is initiated a search of the table reference numbers is performed to determine the TDATA array location of the table data. Once the initial TDATA array location is determined, the function routine SPLNQ1

is used repeatedly to interpolate and/or extrapolate the table data. The first 30 array locations in the TDATA array are reserved for values of table reference numbers and the next 30 array locations are reserved for values indicating the starting location of each table data set. The TDATA storage locations starting at 61 are reserved for the actual table data in groups of 3n+3 storage locations as illustrated in the example which follows.

Example

In Figure 1 a three independent variables table look-up function is shown. The first, second and third independent variables are respectively X, Y, and Z. In this case Z represents a plane and X and Y represent independent variables lying on any plane, Z = constant. The dependent variable is FXYZ. The object of the interpolation and/or extrapolation is to determine FXYZ from any set of input values X_i, Y_i, and Z_i. In this example the TDATA array would contain the following 3n+3 groups:

Z data
Y data
X data
FXYZ data
X data
FXYZ data
.
.
.
.
Y data
X data
FXYZ data
X data
FXYZ data
.
.
.
Y data
X data
FXYZ data
X data

The final value of a three independent variable interpolation is obtained after three steps. The first step is to interpolate or extrapolate (in each constant Z plane and along each constant Y line) in terms of the first independent variable X. This step yields FXYZ for the given value of X_i. The second step is to interpolate or extrapolate (in each constant Z plane) the values of FXYZ (for the given values of X_i) in terms of the second independent variable Y_i. This step yields FXYZ for the given value of X_i and for the given value of Y_i. The final step is merely a single interpolation or extrapolation of 3n+3 sets of data stored in a single dimension array in terms of the third independent variable Z_i.

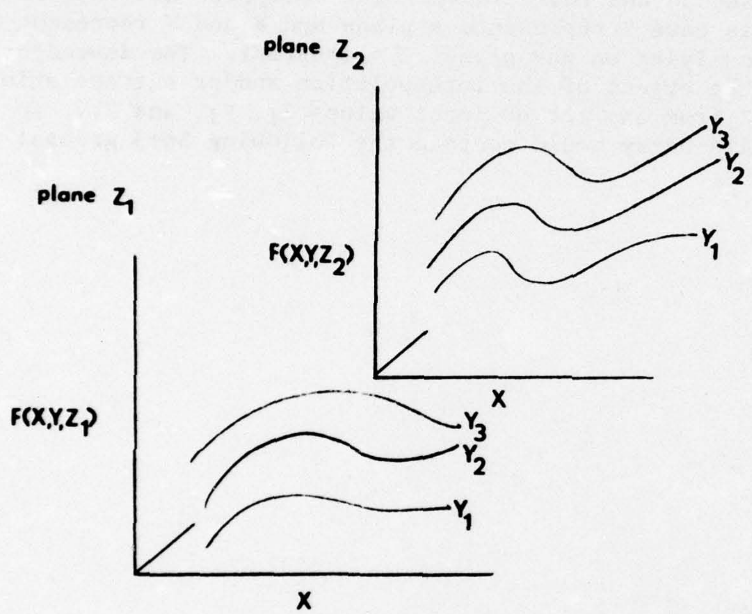

FIGURE 1. THREE INDEPENDENT VARIABLE TABLE LOOK-UP FUNCTION

In each set of 3n+3 storage locations the following information is stored:

1. the number of data points, n
2. the ascending order values of the independent variable at each data point
3. the corresponding values of the dependent variable at each data point
4. a value indicating which two adjacent data points were used in the last interpolation
5. a value indicating either the last calculated second derivatives can be used again or new second derivatives must be calculated
6. n storage locations reserved for the computed second derivatives at each data point

A FORTRAN listing and brief description of the calculation sequence in TREAD/TLOOK is found in Appendix D.

SPLINE Interpolation Technique

An efficient method of interpolating is to plot data points and draw a smooth curve through these using a draftsman's spline. The interpolation is then performed by reading values of the dependent variable at the appropriate point along the smooth drawn curve. This same spline interpolation technique is readily programmed on a computer. Basically, sections of a third degree polynominal are used to connect each pair of adjacent data points in a manner such that the first and second derivatives are continuous at each input data point. The resulting spline curve is not only smooth and continuous, but also passes through each input data point. An energy analogy of the spline curve is as follows: An infinitely thin draftsman's spline is contrained to pass through each data point in a manner such that the strain energy of the spline is a minimum. The computer solution of the spline fit equation involves solving a set of n equations with n unknowns using a matrix elimination method, where n is the number of data points. The n unknowns determined in this method are the second derivatives of the spline curve at each respective data point. After the second derivatives are found a search is performed to determine between which two adjacent data points the given independent variable lies. The interpolated value of the dependent variable for the given independent variable value is then computed using the following equations:

$$y = C_1 * b_k{}^3 + C_2 * (X - X_k)^3 + C_3 * b_k + C_4 * (X - X_k)$$

where: y = interpolated value or dependent variable

X = given value of independent variable

k = subscript of data point with value less than X (determined from search)

$$b_k = X_{k+1} - X$$

$$d_k = X_{k+1} - X_k$$

$$C_1 = y_k'' / 6\ d_k$$

$$C_2 = y_{k+1}''/6d_k$$

$$C_3 = y_k/d_k - y_k'' d_k/6$$

$$C_4 = y_{k+1}/d_k - y_{k+1}''\ d_k/6$$

A more complete derivation of the cubic spline fit equations is found in reference (b). The calculations for the interpolated value are lengthy and complex and require computations of second derivatives. For this reason considerable computation time is saved if the second derivatives are saved for the next interpolation. These second derivatives may be used again only if the values of the input data points do not change. Provisions for saving these second derivatives are included in SPLNQ1 and represents an improvement over the programming technique of reference (b).

The spline fit curve can give very accurate values for first and second derivatives, providing the spline points are accurately given. The derivatives are very sensitive to small errors in coordinates when points are close together. The best procedure is to use as few points as is reasonable. The point spacing should be proportional to the curvature and to the slope of the fitted function. With a very sharp bend or with a steep slope near vertical within a few degrees, spline points should be quite close and very accurate.

A FORTRAN listing and brief description of the calculation sequence in SPLNQ1 is found in Appendix E.

R E C O M M E N D A T I O N S

It is recommended that TREAD/TLOOK be used where applicable in all future computer code developments. It is further recommended that the TREAD/TLOOK routine package be gradually introduced into existing computer codes. This will permit compatible table data inputs between the various operating programs as required.

R E F E R E N C E S

(a) Caddy, M. J., NADC Report No. NADC-76367-30, GPPR - A Multipurpose Computer Code for Data Plotting, unpublished

(b) Pennington, R. H., 1970, "Introductory Computer Methods and Numerical Analysis", the MacMillan Company, London

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

USER'S GUIDE

APPENDIX A
USER'S GUIDE

Subroutine TREAD (with entry TLOOK) is capable of accepting up to 30 table look up functions. The data for these table look up functions is read from input data cards into the subroutine. The input data read-in mode is initiated by the user's program with a FORTRAN statement as follows:

```
CALL TREAD (II, X, Y, Z, FXYZ)
```

Only the parameter, II, need be defined in the input data read-in mode. If II = 1 table input data will be output. If II = 0 table input data will not be output. The table input data card deck consist of groups of cards, each group representing the inputs to a single table look-up function. Each group of cards representing a single table look-up function is composed of the following cards:

Card	Item	Variable Name	Format	Definition
1	1	ITABNO	I, (columns 2-5)	table reference number of table look-up function. If ITABNO = 0 the table data read-in mode is terminated.

Card	Item	Variable Name	Format	Definition
1	2	ALPHA/ NUMERIC	(columns 6 to 75)	table title identification

Card	Item	Variable Name	Format	Definition
1	3	NPC	I5 (columns 76-80)	table input card format switch. If NPC = 0 the card input format is (A4, I3, 7X, 8F7.0) If NPC = 1 the card input format is (A4, I3, 3X, 7F10.0) (see card 2 item 3 and on)

Card	Item	Variable	Format	Definition
2	1	ID	A4 (columns 1-4)	ID is a 4 character identifier used to identify the third independent variable. If table look-up is 2 degree or less use a dummy identifier.

Card	Item	Variable	Format	Definition
2	2	N	I3 (columns 5-7)	N is the number of values to be read in on the remainder of this card and other cards if necessary. (N must be less than 100)

Card	Items	Variable	Format	Definition
2	3 and on	A	8F7.0 (columns 15-21, 22-28, etc.) or 7F10.0 (Columns 11-20, 21-30, etc.) see item 3 on card 1	These are the values of third independent variable in ascending order.

Note: If the value of N is greater than 7 (when NPC = 0) or greater than 8 (when NPC = 1) the remaining values of the third independent variable (not included on the previous card) are input using either the format (14X, 8F7.0) or 10X, 7F10.0) according to the value of NPC, item 3 card 1.

Card	Item	Variable	Format	Definition
3 and following	(All remaining cards have the same format as card 2. The item which distinguishes the card types is the value of the ID variable. The 4 character values of each ID variable on the first four cards must not be identical. The 4 characters of each ID variable on the four cards (after the title card) are user selected. The card order of each ID variable is significant. The first four cards with respective ID variable are as follows:			

Card*	Definition
2	third independent variable, ID and values
3	second independent variable, ID and values
4	first independent variable, ID and values
5	dependent variable, ID and values

The remaining input cards use these same ID values as input above. On cards 4 and 5 are the dependent and first independent variable values along the line given by the first value of the second independent variable and in the plane of the first value of the third independent variable.

Cards with the same respective ID value as cards 4 and 5 are repeated for different values of second independent variable until all second independent variables have been exhausted. The next card has an ID value corresponding to the second independent variable and new values of that variable for the plane of the second value of third independent variable. The values of the first independent variable need not be repeated if they are the same along each second independent equals constant line. In each instance where the values are changed a new card is required. Each table is ended by the ID variable

*(after title card)

with characters set to EOT (end of table). An example of a symmetrical table and unsymmetrical table is illustrated in the sample problem, Appendix B.

APPENDIX B

SAMPLE PROBLEMS

APPENDIX B
SAMPLE PROBLEMS

In Appendix B the use of subroutine TREAD/TLOOK will be illustrated through a sample problem with two table data sets.

Table Input Data Preparation

Part (a) Unsymmetrical Example

A french curve was used to arbitrarily draw the three curves shown in Figure B-1. Data points were obtained from these curves at the points indicated by the symbols on Figure B-1. These data were then input into the TREAD/TLOOK subroutine. The input data cards for this table are shown in Table B-I, cards 1 to 10. On card 1, the table reference number and table identification are shown. On card 2, the third independent variable identifier, number of values and values of the variable are shown. In this table, the third independent variable is not used and therefore Z is a dummy identifier. On card 3 the information for the second independent variable is shown. As indicated on card 3, ARG2 (the second independent variable) has 3 values: 1., 2., and 3., respectively. On cards 4 to 9, the values of the first independent variable (ARG1), and respective values of the dependent variable (FUN) are shown. For example, for the curve ARG2 = 1. shown in Figure B-1, five values of ARG1 and FUN are input on cards 4 and 5, respectively. Other values for ARG2 = 2. and ARG3 are input on lines 6 through 8, respectively.

Part (b) Symmetrical Example

On cards 11 to 32 input cards to a second table are shown. This example could represent an aircraft drag polar which is a function of Mach number and sweep angle. On card 11 the number 1 is shown in column 80 of the input card indicating the optional input data format is used. For this table SWEP, MACH, and CL are the third, second, and first independent variables, respectively, and CD is the dependent variable. On card 21, CL values are input and retained as the first independent variable values for all of the remaining CD values. The CL values are retained because the CL values are not redefined in the remaining data cards. This illustrated the symmetrical feature of the input scheme. The MACH values for the second value of the third independent variable are shown on card 28. As previously described in the user's guide, Appendix A, the second independent variable values (MACH in this example) must be repeated in both symmetrical and unsymmetrical table inputs.

The computer output for a sample problem using TREAD/TLOOK is shown in Appendix C. Both of the previously described tables are input but only the first table (table reference number 263) is used. The main program used to implement this example is not shown. The main program only calls TREAD to input the data and TLOOK to look-up the dependent

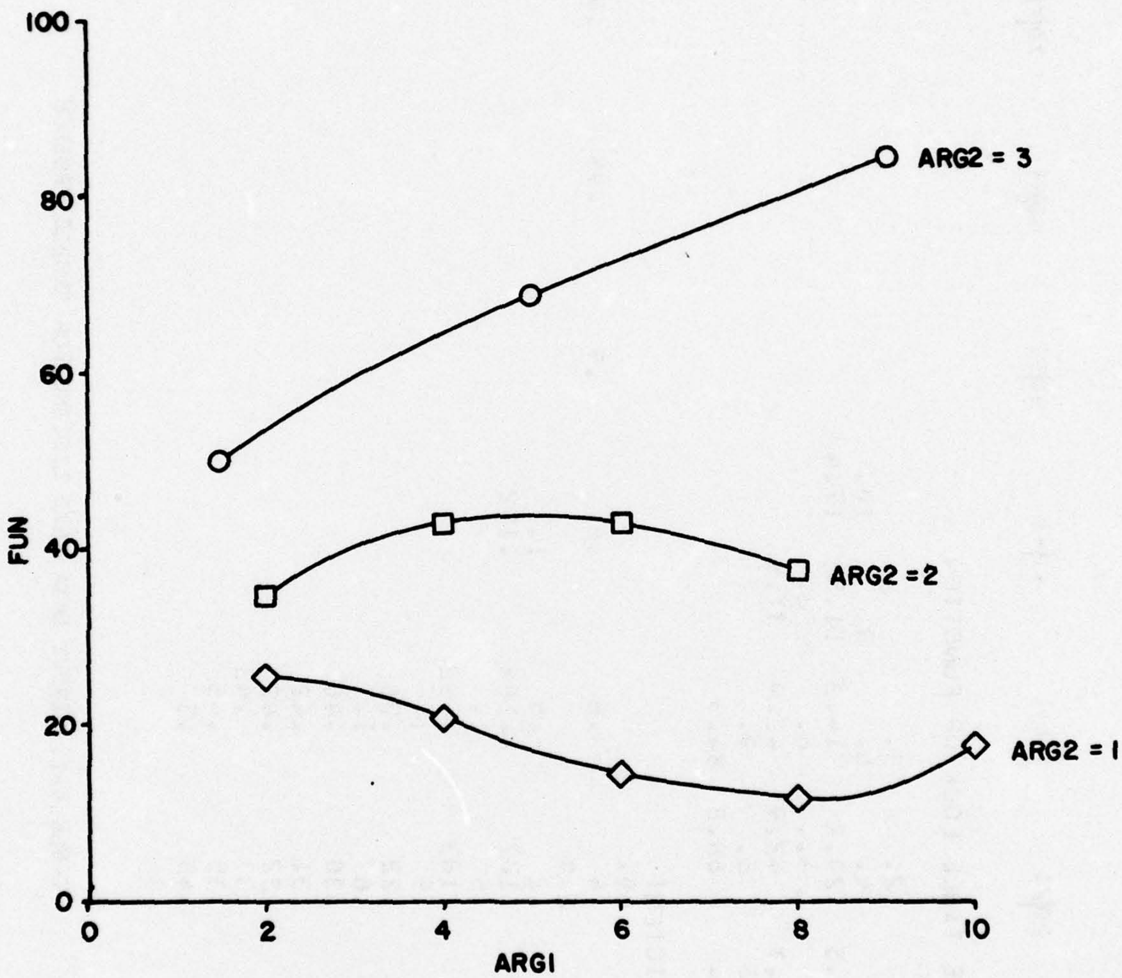

FIGURE B-1. FUNCTION REPRESENTATION FOR SAMPLE PROBLEM

```
1|      10|11      20|21      30|31      40|41      50|51      60|61      70|71      80|  Column
                                                                                          CARD
  263   TEST CASE TABLE LOOK UP FUNCTION                                                     1
Z      1        0.                                                                           2
ARG2   3        1.       2.       3.                                                         3
ARG1   5        2.       4.       6.       8.       10.                                      4
FUN    5        25.5     20.8     14.3     11.5     17.4                                     5
ARG1   4        2.       4.       6.       8.                                                6
FUN    4        34.7     42.9     42.9     37.5                                              7
ARG1   3        1.5      5.       9.                                                         8
FUN    3        50.      68.8     84.3                                                       9
EOT                                                                                         10
 1001 DRAG COEFFICIENT                                                                 1    11
SWEP   2   20.           60.                                                                12
MACH   9    .2           .4           .6          .8          .9         .95        .97     13
           1.            2.0                                                                14
CL     4   0.            .2           .6          1.                                        15
CD     4   .18           .182         .186        .192                                      16
CL     3   0.            .5           1.                                                    17
CD     3   .18           .183         .192                                                  18
CL     3   .12           .5           1.                                                    19
CD     3   .20           .22          .30                                                   20
CL     3   .0            .6           1.                                                    21
CD     3   .26           .30          .40                                                   22
CD     3   .27           .31          .42                                                   23
CD     3   .28           .32          .42                                                   24
CD     3   .29           .33          .43                                                   25
CD     3   .31           .35          .45                                                   26
CD     3   .41           .45          .5                                                    27
```

TABLE B-I. INPUT DATA CARD LISTING FOR SAMPLE PROBLEM

Column 1	10	11	20	21	30	31	40	41	50	51	60	61	70	71	80	CARD
MACH	3	.2		.6		.8										28
CD	3	.15		.151		.154										29
CD	3	.18		.19		.23										30
CD	3	.21		.23		.26										31
EOT																32

TABLE B-I. CONTINUED

variable values. The computer output resulting when TREAD is called from the main program is shown on pages C-2 through C-5, Appendix C. On pages C-2 through C-4 the table input data are listed as processed from the input cards. On page C-5 a table data summary is shown. The purpose of this summary is to show the user how many tables were input, what each table reference number is, the starting array location of each table, and table storage locations used and locations remaining. Computer output from a main program using repeated TLOOK calls to table reference number 263 is shown on pages C-6 through C-8. These data are plotted along with the original table input data points in Figure B-2. As shown in Figure B-2 the interpolated and extrapolated values follow the curve very satisfactorily.

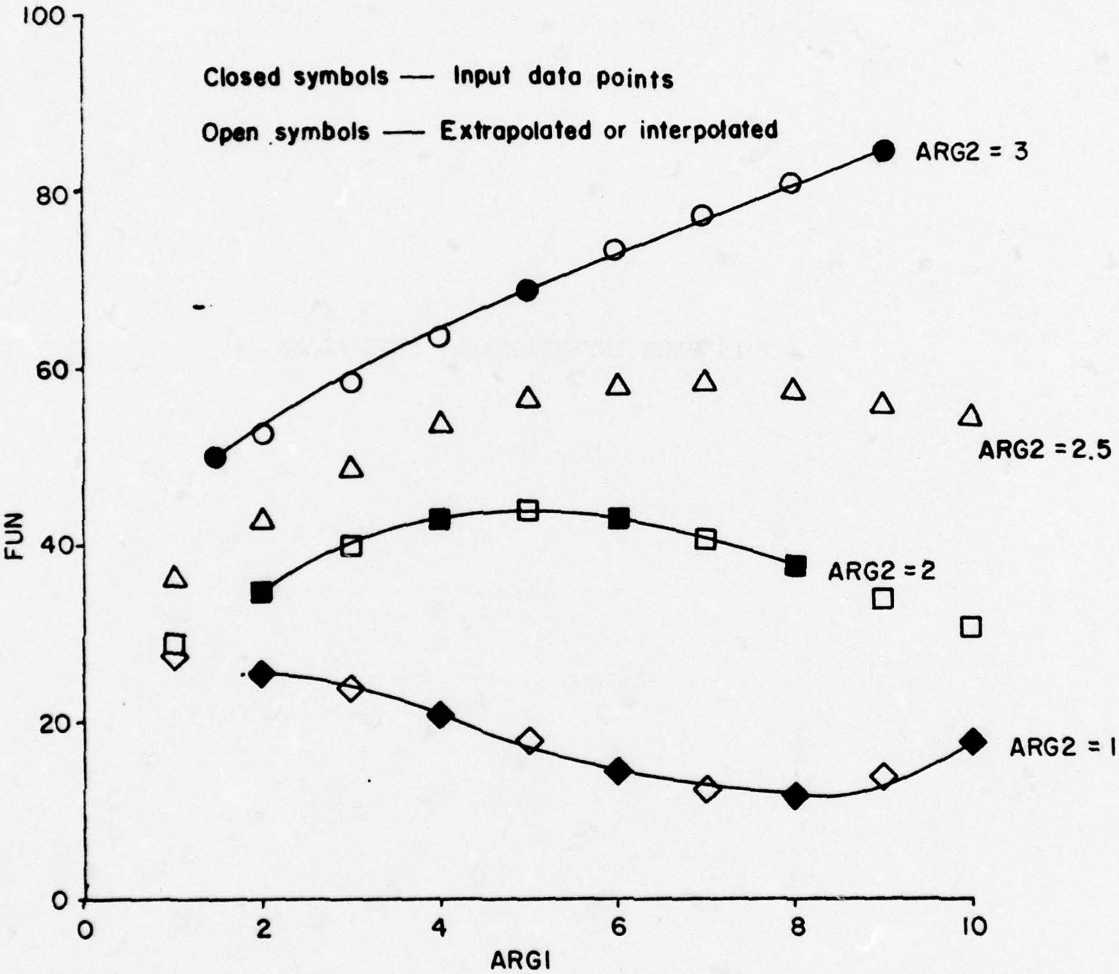

FIGURE B-2. COMPARISON OF INPUT DATA AND INTERPOLATED DATA POINTS

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

COMPUTER OUTPUT FOR SAMPLE PROBLEM

263 TEST CASE TABLE LOOK UP FUNCTION

Z	=	0.	ARG2=	.10000E+01				
			ARG1	.20000E+01	.40000E+01	.60000E+01	.80000E+01	.10000E+02
			FUN	.25500E+02	.20800E+02	.14300E+02	.11500E+02	.17400E+02
Z	=	0.	ARG2=	.20000E+01				
			ARG1	.20000E+01	.40000E+01	.60000E+01	.80000E+01	
			FUN	.34700E+02	.42900E+02	.42900E+02	.37500E+02	
Z	=	0.	ARG2=	.30000E+01				
			ARG1	.15000E+01	.50000E+01	.90000E+01		
			FUN	.50000E+02	.68800E+02	.84300E+02		

1001 DRAG COEFFICIENT

```
SWEP=   .20000E+02 MACH=   .20000E+00
                   CL     0.              .20000E+00   .60000E+00   .10000E+01
                   CD      .18000E+00     .18200E+00   .18600E+00   .19200E+00
SWEP=   .20000E+02 MACH=   .40000E+00
                   CL     0.              .50000E+00   .10000E+01
                   CD      .18000E+00     .18300E+00   .19200E+00
SWEP=   .20000E+02 MACH=   .60000E+00
                   CL      .12000E+00     .50000E+00   .10000E+01
                   CD      .20000E+00     .22000E+00   .30000E+00
SWEP=   .20000E+02 MACH=   .80000E+00
                   CL     0.              .60000E+00   .10000E+01
                   CD      .26000E+00     .30000E+00   .40000E+00
SWEP=   .20000E+02 MACH=   .90000E+00
                   CL     0.              .60000E+00   .10000E+01
                   CD      .27000E+00     .31000E+00   .42000E+00
```

```
SWEP=   .20000E+02 MACH=  .95000E+00
                   CL    0.           .60000E+00   .10000E+01
                   CD     .28000E+00  .32000E+00   .42000E+00
SWEP=   .20000E+02 MACH=  .97000E+00
                   CL    0.           .60000E+00   .10000E+01
                   CD     .29000E+00  .33000E+00   .43000E+00
SWEP=   .20000E+02 MACH=  .10000E+01
                   CL    0.           .60000E+00   .10000E+01
                   CD     .31000E+00  .35000E+00   .45000E+00
SWEP=   .20000E+02 MACH=  .20000E+01
                   CL    0.           .60000E+00   .10000E+01
                   CD     .41000E+00  .45000E+00   .50000E+00
SWEP=   .60000E+02 MACH=  .20000E+00
                   CL    0.           .60000E+00   .10000E+01
                   CD     .15000E+00  .15100E+00   .15400E+00
SWEP=   .60000E+02 MACH=  .60000E+00
                   CL    0.           .60000E+00   .10000E+01
                   CD     .18000E+00  .19000E+00   .23000E+00
SWEP=   .60000E+02 MACH=  .80000E+00
                   CL    0.           .60000E+00   .10000E+01
                   CD     .21000E+00  .23000E+00   .26000E+00
```

TABLE DATA INPUT SUMMARY 2 TABLES

TABLE NUMBER	REFERENCE NUMBER	ARRAY LOCATION
1	263.	61.
2	1001.	124.

DATA STORAGE ALLOCATION 2999
DATA STORAGE NOT USED 2677

```
ARG1=    .10000E+01 ARG2=    .10000E+01 FUN=    .27315E+02
ARG1=    .20000E+01 ARG2=    .10000E+01 FUN=    .25500E+02
ARG1=    .30000E+01 ARG2=    .10000E+01 FUN=    .23451E+02
ARG1=    .40000E+01 ARG2=    .10000E+01 FUN=    .20800E+02
ARG1=    .50000E+01 ARG2=    .10000E+01 FUN=    .17522E+02
ARG1=    .60000E+01 ARG2=    .10000E+01 FUN=    .14300E+02
ARG1=    .70000E+01 ARG2=    .10000E+01 FUN=    .11997E+02
ARG1=    .80000E+01 ARG2=    .10000E+01 FUN=    .11500E+02
ARG1=    .90000E+01 ARG2=    .10000E+01 FUN=    .13439E+02
ARG1=    .10000E+02 ARG2=    .10000E+01 FUN=    .17400E+02
ARG1=    .10000E+01 ARG2=    .15000E+01 FUN=    .26292E+02
ARG1=    .20000E+01 ARG2=    .15000E+01 FUN=    .29084E+02
ARG1=    .30000E+01 ARG2=    .15000E+01 FUN=    .31294E+02
ARG1=    .40000E+01 ARG2=    .15000E+01 FUN=    .31974E+02
ARG1=    .50000E+01 ARG2=    .15000E+01 FUN=    .30825E+02
ARG1=    .60000E+01 ARG2=    .15000E+01 FUN=    .28404E+02
ARG1=    .70000E+01 ARG2=    .15000E+01 FUN=    .25442E+02
ARG1=    .80000E+01 ARG2=    .15000E+01 FUN=    .22539E+02
ARG1=    .90000E+01 ARG2=    .15000E+01 FUN=    .20347E+02
ARG1=    .10000E+02 ARG2=    .15000E+01 FUN=    .18954E+02
ARG1=    .10000E+01 ARG2=    .20000E+01 FUN=    .28964E+02
ARG1=    .20000E+01 ARG2=    .20000E+01 FUN=    .34700E+02
ARG1=    .30000E+01 ARG2=    .20000E+01 FUN=    .39720E+02
ARG1=    .40000E+01 ARG2=    .20000E+01 FUN=    .42900E+02
```

```
ARG1=    .50000E+01 ARG2=    .20000E+01 FUN=    .43827E+02
ARG1=    .60000E+01 ARG2=    .20000E+01 FUN=    .42900E+02
ARG1=    .70000E+01 ARG2=    .20000E+01 FUN=    .40670E+02
ARG1=    .80000E+01 ARG2=    .20000E+01 FUN=    .37500E+02
ARG1=    .90000E+01 ARG2=    .20000E+01 FUN=    .33964E+02
ARG1=    .10000E+02 ARG2=    .20000E+01 FUN=    .30427E+02
ARG1=    .10000E+01 ARG2=    .25000E+01 FUN=    .36152E+02
ARG1=    .20000E+01 ARG2=    .25000E+01 FUN=    .42801E+02
ARG1=    .30000E+01 ARG2=    .25000E+01 FUN=    .48862E+02
ARG1=    .40000E+01 ARG2=    .25000E+01 FUN=    .53523E+02
ARG1=    .50000E+01 ARG2=    .25000E+01 FUN=    .56464E+02
ARG1=    .60000E+01 ARG2=    .25000E+01 FUN=    .57876E+02
ARG1=    .70000E+01 ARG2=    .25000E+01 FUN=    .58078E+02
ARG1=    .80000E+01 ARG2=    .25000E+01 FUN=    .57256E+02
ARG1=    .90000E+01 ARG2=    .25000E+01 FUN=    .55778E+02
ARG1=    .10000E+02 ARG2=    .25000E+01 FUN=    .54023E+02
ARG1=    .10000E+01 ARG2=    .30000E+01 FUN=    .47035E+02
ARG1=    .20000E+01 ARG2=    .30000E+01 FUN=    .52934E+02
ARG1=    .30000E+01 ARG2=    .30000E+01 FUN=    .58587E+02
ARG1=    .40000E+01 ARG2=    .30000E+01 FUN=    .63899E+02
ARG1=    .50000E+01 ARG2=    .30000E+01 FUN=    .68800E+02
ARG1=    .60000E+01 ARG2=    .30000E+01 FUN=    .73244E+02
ARG1=    .70000E+01 ARG2=    .30000E+01 FUN=    .77268E+02
ARG1=    .80000E+01 ARG2=    .30000E+01 FUN=    .80934E+02
ARG1=    .90000E+01 ARG2=    .30000E+01 FUN=    .84300E+02
ARG1=    .10000E+02 ARG2=    .30000E+01 FUN=    .87537E+02
ARG1=    .10000E+01 ARG2=    .35000E+01 FUN=    .59355E+02
```

```
ARG1=   .20000E+01 ARG2=   .35000E+01 FUN=   .63857E+02
ARG1=   .30000E+01 ARG2=   .35000E+01 FUN=   .68540E+02
ARG1=   .40000E+01 ARG2=   .35000E+01 FUN=   .74178E+02
ARG1=   .50000E+01 ARG2=   .35000E+01 FUN=   .81020E+02
ARG1=   .60000E+01 ARG2=   .35000E+01 FUN=   .88764E+02
ARG1=   .70000E+01 ARG2=   .35000E+01 FUN=   .97152E+02
ARG1=   .80000E+01 ARG2=   .35000E+01 FUN=   .10614E+03
ARG1=   .90000E+01 ARG2=   .35000E+01 FUN=   .11543E+03
ARG1=   .10000E+02 ARG2=   .35000E+01 FUN=   .12491E+03
```

APPENDIX D

TREAD/TLOOK
DESCRIPTION AND FORTRAN LISTING

APPENDIX D
TREAD/TLOOK
DESCRIPTION AND FORTRAN LISTING

DESCRIPTION OF TREAD/TLOOK CALCULATION SEQUENCE

The input section of TREAD/TLOOK is shown on cards 14 to 121, and the interpolation section is shown on cards 122 to 158, Table D-I. In the first part of the interpolation section, a search is performed to find the correct table location in the TDATA array to begin operating on the data. On cards 146 to 156 the three interpolation or extrapolations (one for each independent variable) are performed using the SPLNQ1 routine. The final dependent variable value returned is shown on card 156.

```
      SUBROUTINE  TREAD (II,X,Y,Z,FXYZ)
C*** G E N E R A L   P U R P O S E   I N T E R P O L A T I O N   R O U T I
C*** M CADDY    6/71
      DIMENSION       TDATA( 2999)             ,A(100),IDD(4),IFM(4,2)
      DATA IFM/10H(A4,I3,7X,,10H8F7.0)      ,10H(14X8F7.0 ,10H)
     1        ,10H(A4,I3,3X,,10H7F10.0)    ,10H(10X7F10.0,10H)           /
      LOC=61
      ITPRT=II
      NTBL=0
      NMAX=1200
      DO 10 M=1,NMAX
   10 TDATA(M)=0.
      ILAST=1
   20 READ 30, IP,ITABNO,NPC
   30 FORMAT (I1,I4,70H
     1                        ,I5)
      IF (ITABNO) 120,40,120
   40 NREM=NMAX-LOC
      IF(NTBL.LE.1) GO TO 70
C*** RE ORDER TABLE REFERENCE NUMBERS
   50 NDONE=1
      DO 60 L=2,NTBL
      K1=L-1
      K2=L
      IF(TDATA(K1).LT.TDATA(K2)) GO TO 60
      XSAVE=TDATA(K1)
      TDATA(K1)=TDATA(K2)
```

TABLE D-I. FORTRAN LISTING OF TREAD/TLOOK

```
      TDATA(K2)=XSAVE
      XSAVE=TDATA(K1+30)
      TDATA(K1+30)=TDATA(K2+30)
      TDATA(K2+30)=XSAVE
      NDONE=0
   60 CONTINUE
      IF(NDONE.EQ.0) GO TO 50
   70 IF(II)440,440,80
   80 PRINT 90,NTBL
   90 FORMAT (1H1,35X,24HTABLE DATA INPUT SUMMARY,1XI2,1X,6HTABLES,//,28
     1X,48HTABLE NUMBER  REFERENCE NUMBER   ARRAY LOCATION )
      PRINT 100,((N,TDATA(N),TDATA(N+30)),N=1,NTBL)
  100 FORMAT (32X,I2,12X,F5.0,15X,F5.0)
      PRINT 110, NMAX,NREM
  110 FORMAT (//,35X,24HDATA STORAGE ALLOCATION ,I4,/35X,24HDATA STORAGE
     1 NOT USED    ,I4,/)
C     PRINT 1111,((M,TDATA(M)),M=1, LOC)
C1111 FORMAT(5(1X,I4,1X,F12.5))
      RETURN
  120 IF (ITPRT) 150,150,130
  130 IP=1
      PRINT 30, IP,ITARNO
      PRINT 140
  140 FORMAT(//////)
  150 CONTINUE
  160 NTBL=NTBL+1
      TDATA(NTBL)=ITARNO
```

TABLE D-I. CONTINUED

```
      TDATA(NTRL+30)=LOC
      ICC=0
      IPP=NPC+1
      NCC=8-NPC
      NCR=NCC+1
      LZ=LOC
      ID=4H
      DO 139 ICL=1,4
  139 IDD(ICL)=4H
  170 IDL=ID
      READ IFM(1,IPP),ID,N,(A(I),I=1,NCC)
      IF(ID.EQ.4HEOT )20,180
  180 IF(N.GT.NCC)190,200
  190 READ IFM(3,IPP),(A(I),I=NCR,N)
  200 ICC=ICC+1
      IF(ICC.LE.4) IDD(ICC)=ID
      IF(ID.EQ.IDL)320,210
  210 IF(ID.EQ.IDD(4))220,240
  220 IF(IDL.EQ.IDD(2))320,230
  230 LOC=LF
      L=LX
      GO TO 260
  240 TDATA(LOC)=N
      L=LOC
      IF(ID.EQ.IDD(3)) LX=L
      IF(ID.EQ.IDD(2))250,300
  250 LY=L
```

TABLE D-I. CONTINUED

```
      L7=L7+1
      GO TO 300
  260 IF(IP.EQ.0) GO TO 300
      LE=LX+N
      LF=LX+1
      LY=LY+1
      PRINT 270,IDD(1),TDATA(L7),IDD(2),TDATA(LY)
  270 FORMAT(1XA4,1H=,E13.5,1XA4,1H=,E13.5)
      JF=0
      M=N
      LF=LX
  280 NP=M
      IF(M.GT.8)NP=8
      M=M-NP
      LE=LF+NP
      LF=LF+1
      PRINT 290,IDD(3),(TDATA(I),I=LF,LE)
  290 FORMAT(20X,A4,1X,8E13.5)
      LF=LE
      JE=JF+NP
      JF=JF+1
      PRINT 290,IDD(4),(A(I),I=JF,JE)
      JF=JE
      IF(M.GT.0) GO TO 280
  300 DO 310 I=1,N
      LOC=LOC+1
  310 TDATA(LOC)=A(I)
```

TABLE D-I. CONTINUED

```
      LF=LOC
      IF(ID.EQ.IDD(4)) TDATA(LOC+2)=1.
      LOC=L+3*N+3
      IF(LOC.GT.NMAX)340,170
  320 TDATA(LOC)=TDATA(LX)
      L=LOC
      DO 330 I=1,N
      LOC=LOC+1
  330 TDATA(LOC)=TDATA(LX+I)
      GO TO 260
  340 PRINT 350, ITABNO
  350 FORMAT (1X,32H*********TABLE OVER FLOW ,TABLE ,I5,11H NOT LOADED)
      GO TO 40
      ENTRY TLOOK
C***********************************************************************
      IF(II)440,440,360
C*** SEARCH FOR CORRECT TABLE
  360 NH=NTBL+1
      NL=1
      K=ILAST
      XTAB=II
      KT=0
  370 KT=KT+1
      IF(KT.GT.6) GO TO 440
      IF(XTAB-TDATA(K)) 390,410,380
  380 NL=K
      GO TO 400
```

TABLE D-I. CONTINUED

```
                                                    CARD
390 NH=K                                            136
400 K=(NH-NL)/2+NL                                  137
    GO TO 370                                       138
410 ILAST=K                                         139
    IZ=TDATA(K+30)                                  140
    INZ=TDATA(IZ)                                   141
    IY=3*INZ+IZ+3                                   142
    IX=IY                                           143
    IZ1=IZ+INZ+1                                    144
    IZ2=IZ+2*INZ                                    145
    DO 430 IZS=IZ1,IZ2                              146
    INY=TDATA(IY)                                   147
    IX=3*INY+IX+3                                   148
    IY1=IY+INY+1                                    149
    IY2=IY+2*INY                                    150
    DO 420 IYS=IY1,IY2                              151
    TDATA(IYS)=SPLNQ1(IX,TDATA,X)                   152
420 IX=IX+3*TDATA(IX)+3                             153
    TDATA(IZS)=SPLNQ1(IY,TDATA,Y)                   154
430 IY=IX                                           155
    FXYZ=SPLNQ1(IZ,TDATA,Z)                         156
440 RETURN                                          157
    END                                             158
```

TABLE D-I. CONTINUED

APPENDIX E

FUNCTION SPLNQ1 DESCRIPTION AND FORTRAN LISTING

APPENDIX E
FUNCTION SPLNQ1 DESCRIPTION AND FORTRAN LISTING

DESCRIPTION OF FUNCTION SPLNQ1 CALCULATION SEQUENCE

A brief description of the calculation sequence in SPLNQ1 is as follows: On cards 8-20 of Table E-I, the storage location subscripts relative to the beginning location of the input data in the X array are computed. On cards 22 to 27, the given independent variable value, XIN, is computed with the low and high value of the input independent variable data point set. If XIN is greater than the highest data point or lower than the lowest data point, the variable NTRAP is set to 0 or 1, respectively. The NTRAP variable is used as a cue for extrapolation. The variable L, on card 30, is the storage location of the upper data point used in the previous interpolation. L will be assigned various values during execution of the routine. The value of L is initialized zero in the X array. In the situation when L is equal to or less than zero a full search (cards 34 to 39) of the independent variable data point set is performed in order to find the correct interval for the interpolation. An optimum search technique is used such that for n data points, only $\log_2(n)$ comparisons need be made to find the interval $X(L - 1) \leq XIN \leq X(L)$.

In the case of large arrays, 50 points for example, a maximum of 6 comparisons need be made to find the interval $X(L - 1) \leq XIN \leq X(L)$.

In the second situation where L is greater than zero, the search is begun by first checking the data point interval used in the last interpolation. After the correct interval is found, the calculation sequence is at cards 42 and 43. The variable M which satisfies $X(M - 1) \leq XIN \leq X(M)$ is stored in the X array for the next interpolation as shown on card 43. On card 46, the values of the variables L and IQMODE are compared with zero. IQMODE is a user set variable which determines whether or not the second derivatives are to be saved and used in the next interpolation sequence. When L = 0 these data points are being interpolated for the first time, therefore, the second derivates must be calculated. When IQMODE = 0 these data points are being interpolated for the first time, therefore, the second derivates must be calculated.

The condition IQMODE = 0 implies that in each usage the input data points to **SPLNQ1** are changing and the second derivates must be recalculated for each interpolation. On cards 47 to 86, the second derivative calculations are performed. The second derivative calculations are optimized to the extent that if IQMODE = 0 only the second derivatives required for the current interpolation are computed.

The calculations on cards 87 to 96, concern extrapolation. A linear extrapolation is performed using the slope at each respective end point of the spline curve. The linear extrapolation technique is preferred rather than a second order extrapolation because far reaching extrapolations are more controlled and predictable.

```
      FUNCTION  SPLNQ1    (NLOC,X,XINDEP)
C     CUBIC SPLINE FIT REVISED 10/21/71 M CADDY
C     THIS VERSION HAS QAO OPTION WHERE ALL OF THE SPLINE COEFFICIENTS
C     ARE COMPUTED AND STORED IN THE ARRAY.  FOR N DATA POINTS 3*N+3
C     STORAGE LOCATIONS ARE REQUIRED FOR THE DATA AND THE COEFFICIENTS
C     NEW FEATURE IS QUICK LOOK-UP FOR LARGE ARRAYS
      DIMENSION G(100),SB(100),X(1)
      XIN=XINDEP
      NS=NLOC
      NOPTS=X(NS)
      ID=NS+NOPTS
      NSP1=NS+1
      NSP2=NS+2
      NS2=NOPTS*2+NSP1
      L=X(NS2)
      LSC=NS2+1
      IQMODE=X(LSC)
      K=L
      NL=NSP1
      NH=ID
      NTRAP=-1
      IF(NOPTS-1)130,130,10
   10 IF(XIN-X(ID))30,140,20
   20 NTRAP=0
      GO TO 140
   30 IF(XIN-X(NSP1))40,50,60
   40 NTRAP=1
```

TABLE E-I.
FORTRAN LISTING OF SPLNQ1

```
                                                   CARD
 50 K=NSP2                                           28
    GO TO 150                                        29
 60 IF(L)120,120,70                                  30
 70 IF(XIN-X(K))80,150,100                           31
 80 NH=K                                             32
    K=K-1                                            33
 90 IF(XIN-X(K))110,150,100                          34
100 NL=K                                             35
    GO TO 120                                        36
110 NH=K                                             37
120 K=(NH-NL)/2+NL                                   38
    IF(K-NL)90,140,90                                39
130 YOUT=X(NSP2)                                     40
    GO TO 260                                        41
140 K=NH                                             42
150 M=K                                              43
    X(NS2)=M                                         44
    N=M+NOPTS                                        45
    IF(L*IQMODE)160,160,220                          46
160 X2=X(NSP1)                                       47
    X3=X(NSP2)                                       48
    X32=X3-X2                                        49
    Y3=X(ID+2)                                       50
    Y32=Y3-X(ID+1)                                   51
    G(1)=0.                                          52
    SB(1)=-.5                                        53
    N1=NOPTS-1                                       54
```

TABLE E-I. CONTINUED

```
      DO 170 I=2,N1
      J=NSP1+I
      K1=J+NOPTS
      X1=X2
      X2=X3
      X21=X32
      X3=X(J)
      X32=X3-X2
      Y2=Y3
      Y3=X(K1)
      Y21=Y32
      Y32=Y3-Y2
      W=(X3-X1)/3.-X21*SB(I-1)/6.
      SB(I)=X32/(W*6.)
  170 G(I)=(Y32/X32-Y21/X21-X21*G(I-1)/6.)/W
      EM1=G(N1)/(2.+SB(N1))
      IF(L)180,180,190
  180 ID1=NOPTS
      KQAS=NOPTS+LSC
      X(KQAS)=EM1
      GO TO 200
  190 ID1=ID+2-M
  200 DO 210 I=2,ID1
      EM2=EM1
      EM1=G(N1)-SB(N1)*EM2
      X(N1+LSC)=EM1
  210 N1=N1-1
```

TABLE E-I. CONTINUED

```
                                                                      CARD
      IF(L)220,220,230                                                  82
  220 NSM=NS2+M-NS+1                                                    83
      EM1=X(NSM-1)                                                      84
      EM2=X(NSM)                                                        85
  230 S=X(M)-X(M-1)                                                     86
      IF(NTRAP)250,240,240                                              87
  240 IX=M-NTRAP                                                        88
      IY=IX+NOPTS                                                       89
      XS=XIN                                                            90
      XIN=X(IX)                                                         91
      Z1=X(M)-XIN                                                       92
      Z2=XIN-X(M-1)                                                     93
      YOUT=(((EM2*Z2*Z2-EM1*Z1*Z1)/2.+ X(N)-X(N-1))/S                   94
     1-(EM2-EM1)*S/6.)*(XS-XIN)+X(IY)                                   95
      GO TO 260                                                         96
  250 Z2=XIN-X(M-1)                                                     97
      Z1=X(M)-XIN                                                       98
      YOUT=(EM1*Z1*Z1*Z1+EM2*Z2*Z2*Z2)/6./S+(X(N)/S-EM2*S/6.)*Z2        99
     1+(X(N-1)/S-EM1*S/6.)*Z1                                          100
  260 SPLN01=YOUT                                                      101
      RETURN                                                           102
      END                                                              103
```

TABLE E-I. CONTINUED

D I S T R I B U T I O N   L I S T

REPORT NO. NADC-76366-30

AIRTASK NO. A5365360/202F/OW45430000
WORK UNIT NO. OR6-1-24

	No. of Copies
NAVAIR (AIR-954) . . . . (2 for retention) (2 for AIR-503Z)	4
DDC	12
NASA Lewis Research Center, Cleveland, OH 44135. . . . . . (1 for library) (3 for L. H. Fishbach, M. S. 106)	4